Exp:15

Code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_FILES\_PER\_DIR 100

#define MAX\_DIRS 100

struct File {

char name[50];

int size;

};

struct Directory {

char name[50];

struct File files[MAX\_FILES\_PER\_DIR];

int file\_count;

};

struct TwoLevelDir {

struct Directory dirs[MAX\_DIRS];

int dir\_count;

};

void addFile(struct TwoLevelDir \*dir, char \*dir\_name, char \*file\_name, int size) {

int i, found = 0;

for (i = 0; i < dir->dir\_count; i++) {

if (strcmp(dir->dirs[i].name, dir\_name) == 0) {

if (dir->dirs[i].file\_count < MAX\_FILES\_PER\_DIR) {

struct File new\_file;

strcpy(new\_file.name, file\_name);

new\_file.size = size;

dir->dirs[i].files[dir->dirs[i].file\_count++] = new\_file;

printf("File '%s' added to directory '%s'\n", file\_name, dir\_name);

} else {

printf("Directory '%s' is full, cannot add more files\n", dir\_name);

}

found = 1;

break;

}

}

if (!found) {

printf("Directory '%s' not found\n", dir\_name);

}

}

void displayContents(struct TwoLevelDir \*dir) {

int i, j;

for (i = 0; i < dir->dir\_count; i++) {

printf("Directory: %s\n", dir->dirs[i].name);

printf("Files:\n");

for (j = 0; j < dir->dirs[i].file\_count; j++) {

printf(" %s - Size: %d\n", dir->dirs[i].files[j].name, dir->dirs[i].files[j].size);

}

}

}

int main() {

struct TwoLevelDir root\_dir;

root\_dir.dir\_count = 0;

addFile(&root\_dir, "Documents", "Report.docx", 2048);

addFile(&root\_dir, "Documents", "Presentation.pptx", 4096);

addFile(&root\_dir, "Images", "Photo1.jpg", 1024);

displayContents(&root\_dir);

return 0;

}

Output:

